MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA DE UM LABORATÓRIO DO DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELÉTRICA

SOUSA, A. R. Guilherme[1], PADILHA, S. J. Vitor[1], MONTERO, J. E. Ciro[2]

[1]Acadêmico, Curso de Engenharia Elétrica - DAEE, Fundação Universidade Federal de Rondônia, Porto Velho, Rondônia, Brasil (e-mails: guilhermeroqueinfor@gmail.com, padilha.unir@gmail.com, kawanvicentin@gmail.com)

[2]Professor do Curso de Engenharia Elétrica - DAEE, Fundação Universidade Federal de Rondônia, Sala 314 - 4H, Porto Velho, Rondônia, Brasil (e-mail: ciro.egoavil@unir.br)

RESUMO Este projeto tem como objetivo desenvolver um protótipo de medidor de consumo de energia elétrica para monitoramento remoto em um laboratório do Departamento Acadêmico de Engenharia Elétrica da Universidade Federal de Rondônia. Através da integração com o *Firebase*, para possibilitar o acompanhamento em tempo real do consumo energético, por meio de um *website* e o uso de microcontroladores.

PALAVRAS CHAVE Medidor, consumo, monitoramento, *firebase*, *website*, microcontroladores.

I. INTRODUÇÃO

A engenharia elétrica desempenha um papel fundamental na busca por soluções inovadoras que visam à eficiência energética, um dos principais desafios da atualidade. Em ambientes acadêmicos, como os laboratórios universitários, a gestão eficiente do consumo de energia é essencial não apenas para a redução de custos operacionais, mas também para o desenvolvimento de práticas sustentáveis e a promoção de um consumo mais consciente de recursos naturais. Nesse contexto, a aplicação de tecnologias modernas de monitoramento de energia pode contribuir significativamente para otimizar o uso da eletricidade, especialmente quando combinada com ferramentas de automação e comunicação de dados [1].

O projeto propõe o desenvolvimento de um protótipo de medidor de energia elétrica, integrado a uma plataforma de monitoramento remoto que utiliza *Firebase*, que permite a visualização em tempo real dos dados de consumo por meio de um *website*. A utilização de sensores e microcontroladores no protótipo visa garantir a precisão nas medições e a transmissão de dados de forma eficiente, enquanto o *Firebase* proporciona o armazenamento e a atualização dinâmica das informações [2].

A combinação de sistemas de medição de energia com plataformas de *IoT* (Internet das Coisas) tem ganhado destaque na engenharia elétrica, sendo aplicada tanto em projetos acadêmicos quanto em ambientes comerciais e residenciais. A possibilidade de monitoramento remoto e análise de dados em tempo real permite a identificação de padrões de consumo e a implementação de estratégias para a redução de desperdícios, o que contribui diretamente para a melhoria da eficiência energética. Além disso, o uso de ferramentas como o *Firebase* facilita a implementação de soluções escaláveis, permitindo a integração de novos dispositivos ou a ampliação do sistema de monitoramento para outros ambientes [4].

II. JUSTIFICATIVA

A eficiência energética é um dos principais desafios da engenharia elétrica, especialmente em ambientes acadêmicos, onde o consumo de energia muitas vezes não é monitorado de forma eficiente. O desenvolvimento de um sistema de monitoramento remoto de consumo de energia elétrica justifica-se pela necessidade de otimizar o uso de recursos energéticos, reduzir custos operacionais e minimizar o impacto ambiental.

Os sensores de medição em conjunto com a plataforma *Firebase*, permite que o sistema armazene a análise em tempo real dos dados de consumo, que facilita a identificação de padrões e a implementação de estratégias para a redução de desperdícios. Essa abordagem não apenas melhora a gestão energética, mas também contribui para a sustentabilidade do ambiente, promovendo o uso responsável da energia.

III. OBJETIVOS

Para o desenvolvimento do projeto foram considerados os seguintes objetivos:

A. OBJETIVO GERAL

Implementar um sistema de medição de consumo de energia conectado ao *website*.

B. OBJETIVOS ESPECÍFICOS:

- Projetar e implementar o protótipo de um medidor de consumo de energia elétrica capaz de
- monitorar os parâmetros de consumo em tempo real.
- Integrar o sistema de medição com a plataforma *firebase* para armazenamento e atualização dos dados de consumo.
- Desenvolver um *website* interativo que permita a visualização e análise do consumo de energia elétrica de forma remota.
- Validar o desempenho do sistema, garantindo a precisão na medição e a funcionalidade da interface web.
- Analisar os dados coletados para identificar padrões de consumo e oportunidades de otimização da eficiência energética do laboratório.

IV. METODOLOGIA

A. REVISÃO DE LITERATURA

1) Sistemas de Medição de Consumo de energia elétrica

Existem 3 tipos de sistemas de medição de energia elétrica principais, sendo eles:

- *Home Area Network*: Trata-se de uma rede residencial destinada à comunicação entre dispositivos digitais. Atualmente, as *HANs* têm um novo propósito, que é fornecer uma visão do Smart Grid no nível residencial, permitindo que os consumidores compreendam e gerenciem melhor o seu consumo de energia;
- *Home Energy Monitors*: São dispositivos desenvolvidos especificamente para ajudar os consumidores a monitorar e reduzir seu consumo de energia. Além disso, são mais portáteis, simples e acessíveis em comparação aos contadores inteligentes. Esses monitores conseguem medir e processar os consumos elétricos em tempo real, embora não substituam os contadores tradicionais;
- Sistemas para Medição de Consumo de Eletrodomésticos: Refere-se a técnicas que permitem a obtenção de informações detalhadas sobre o consumo de energia de cada eletrodoméstico. Esses sistemas fornecem dados de forma que o consumidor possa facilmente entender, como o valor do consumo em termos monetários, que descomplica o gerenciamento eficiente do uso de energia.

2) *Internet of Things*

A IoT (*"Internet of Things"* ou "Internet das Coisas") é um termo que surgiu em 1999 por Kevin Ashton e está relacionado a capacidade de dispositivos conectados à Internet tem de realizar determinada ação. Esse tipo de interconexão entre objetos pode ser mais um passo para a criação de ambientes inteligentes.

A partir disso, os dispositivos, ou coisas, podem fornecer informações coletadas de ambientes a partir de uso de sensores, monitoramentos de consumo de eletricidade, e contatar outros dispositivos, emitindo relatórios para notificar uma mudança no consumo de energia.

A Internet das Coisas apresenta-se como uma das principais tecnologias emergentes na atualidade, com o objetivo de mudar os ambientes que incorporam o IoT para otimizar a vida das pessoas a partir da sua praticidade e autonomia, que por sua vez, torna os espaços inteligentes.

3) Microcontroladores

Os microcontroladores são dispositivos que integram todos os componentes necessários para o seu funcionamento, sendo alimentados por uma fonte externa. Funcionam como um microcomputador, mas com um foco específico em tarefas de controle e automação, sendo implementados em circuitos integrados (CIs).

Foi utilizado os microcontroladores ESP32 e o PIC16F877A. O ESP32 é um microcontrolador moderno e versátil, conhecido por suas capacidades de conectividade, como Wi-Fi e Bluetooth, o que o torna ideal para aplicações de Internet das Coisas (IoT). Ele é equipado com múltiplos núcleos de processamento, que proporciona alto desempenho para tarefas mais complexas e, ao mesmo tempo, permite uma grande flexibilidade em termos de conectividade e integração com outros dispositivos, conforme Figura 1.

Figura 1: Microcontrolador Esp32 [4].

Por outro lado, o PIC16F877A é um microcontrolador de 8 bits, amplamente utilizado em projetos mais simples e em sistemas embarcados que não demandam conectividade sem fio. Ele é notável pela sua estabilidade e simplicidade, possuindo uma vasta gama de periféricos integrados, o que facilita a implementação de controle de dispositivos em aplicações como automação residencial, controle de motores e sistemas de medição, como pode ser visto na Figura 2.

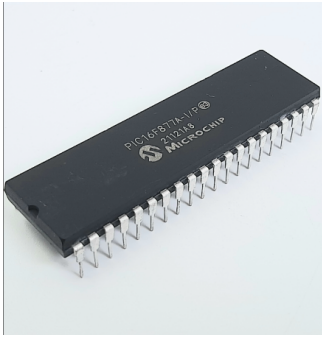



Figura 2: Microcontrolador PIC16F877A [5].

4) Modelo PIC-16F877A

O PIC16F877A (Microchip Technology) é um microcontrolador que se destaca pela sua ampla aplicação em projetos de sistemas embarcados de baixo custo e por seu suporte a diversas interfaces de comunicação, como UART, SPI e I2C. Diferentemente de microcontroladores mais avançados, como o ESP32, o PIC16F877A não possui conectividade *wireless* integrada, exigindo módulos externos para essas funcionalidades.

O PIC16F877A é baseado em uma arquitetura de 8 bits, possuindo um núcleo único. Possui a capacidade de interpretar as linguagens de programação Assembly e C, destacando-se por sua simplicidade e robustez em aplicações onde o desempenho extremo não é necessário. A Tabela 1 apresenta as características do PIC16F877A em comparação com outros controladores da Microchip, que destaca suas vantagens em determinadas situações [5].

Atributo	**Baseline**	**Mid-Range**	**Enhanced Mid-Range**	**PIC18**
Nº de Pinos	6-40	8-64	8-64	18-100
Suporte a Interrupções	sem suporte	Uma interrupção	Uma interrupção com armazenamento de contexto de hardware	Múltiplas interrupções com armazenamento de contexto de hardware
Desempenho	5 MIPS	5 MIPS	8 MIPS	Até 16 MIPS
Instruções	33, 12-bit	35, 14-bit	49, 14-bit	83, 16-bit
Mem. de Programa	até 3 kB	até 14 kB	até 28 kB	até 128 kB
Mem. de Dados	até 134B	até 368B	até 1.5 kB	até 4 kB
Recursos	• Comparador • ADC de 8 bits • Memória de Dados • Oscilador interno	Além dos da Baseline: • SPI/I2C • UART • PWMs • Suporte a LCD • ADC de 10 bits • Op Amp	Além dos da Mid-Range: • Comunicações com múltiplos periféricos • PWMs • Espaço para programação linear • ADC de 10 bits • Temporização independente	Além dos da Enh. Mid-Range: • Multiplicador 8x8 em hardware • CAN • CTMU • USB • Ethernet • ADC de 12 bits
Famílias	PIC10, PIC12, PIC16	PIC12, PIC16	PIC12F1XXX, PIC16F1XXX	PIC18

Tabela 1: PIC 8-bits: Principais Características [5].

5) Sensores de corrente

O sensor de corrente não invasivo STC-013 é amplamente utilizado na engenharia elétrica para medir correntes em circuitos sem a necessidade de desconexão dos condutores. Ele opera como um transformador de corrente tipo "*clip-on*", oferecendo praticidade e segurança na instalação, especialmente em sistemas elétricos de difícil acesso.

Esse sensor é projetado para medir correntes alternadas em cabos e barramentos, com capacidade de detectar correntes de até 100A, dependendo do modelo. A saída do sensor é proporcional à corrente medida, geralmente em forma de tensão ou corrente reduzida, que facilita sua integração com microcontroladores ou dispositivos de monitoramento. O STC-013 é frequentemente empregado em análises de qualidade de energia para identificar problemas como harmônicos, desequilíbrios e sobrecargas, contribuindo para o diagnóstico e melhoria da eficiência energética em sistemas elétricos [6].

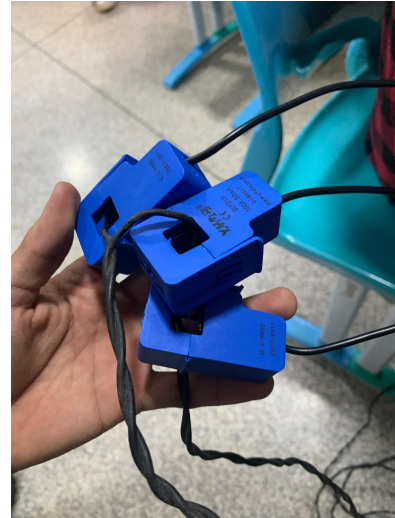



Figura 3: Sensor de corrente STC- 013.

6) Qualidade de Energia do Departamento Acadêmico de Engenharia Elétrica

A Fundação Universidade Federal de Rondônia (UNIR), criada pela Lei 7.011/82, iniciou suas atividades em 1982 com três cursos de Bacharelado (Administração, Ciências Contábeis e Ciências Econômicas), em parceria com a UFPA e vinculada à Prefeitura de Porto Velho, incorporando a FUNDACENTRO. Para atender as necessidades do Estado e ampliar vagas, a UNIR se dedicou à criação de cursos tecnológicos e de formação de profissionais nas Ciências Exatas, como Engenharias. O Departamento Acadêmico de Engenharia Elétrica foi criado em 2007, e o prédio de engenharia foi concluído em 2018.

O prédio do departamento é composto por 4 pavimentos, com os dois primeiros contendo 5 salas de aula, 6 laboratórios e dois banheiros. O terceiro pavimento contém o administrativo e auditório do departamento e o quarto e último pavimento contendo 6 laboratórios, sendo 5 didáticos e 1 de pesquisa.

O departamento apresenta conformidade em relação as normas regulamentadoras da ANEEL de qualidade de energia. O DAEE apresenta uma tensão V nominal monofásica de 127V com poucas oscilações entre 124V e 130V e um fator de potência acima de 0,92 [7].

B. SIMULAÇÃO

Nesta etapa, foi realizada a simulação dos circuitos utilizando o *software* Proteus, com o objetivo de verificar a funcionalidade do circuito antes da aquisição dos componentes físicos e da montagem do protótipo. Essa abordagem permitiu identificar e corrigir possíveis falhas de projeto de forma antecipada, reduzindo custos e otimizando o desenvolvimento.

Além disso, utilizou-se o *software* MikroC para desenvolver e depurar o código do microcontrolador, garantindo a integração adequada entre o hardware e o *firmware*.

Para simular o comportamento dos sensores, foram configurados modelos que representassem o funcionamento dos sensores de corrente STC-013 conectados ao microcontrolador. Três potenciômetros foram implementados no ambiente de simulação, representando os ajustes necessários para calibrar os sensores e definir os limiares de acionamento.

Adicionalmente, o módulo SIM900D foi incorporado ao circuito simulado, viabilizando a conexão *Wi-Fi* para o envio e a atualização de dados de corrente em tempo real. Essa funcionalidade foi essencial para validar a capacidade de comunicação remota do sistema e a integração com uma plataforma *IoT*.

A Figura 4 ilustra a configuração do circuito simulado no *software* Proteus, evidenciando os componentes principais e a interação entre eles.

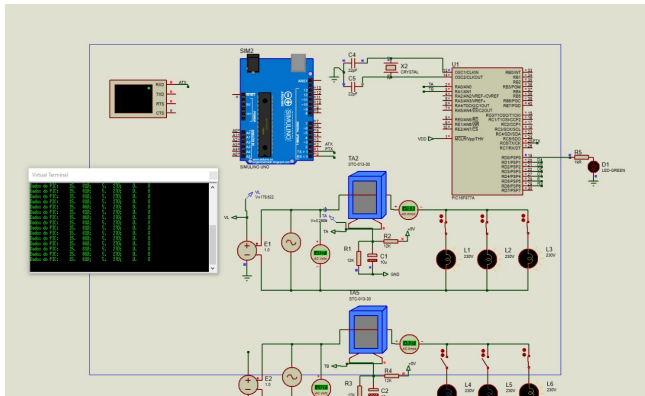



Figura 4: Simulação do projeto.

C. PROGRAMAÇÃO

A programação do pic16f877A foi realizada utilizando o *software* MikroC, onde o fluxograma do projeto é visto na Figura 5. Inicialmente o microcontrolador lê os valores dos 3 sensores analógicos de corrente (utilizando o ADC do PIC), processa os dados, e transmite para o módulo wifi (ESP-32) via UART. O código completo pode ser visto no Apêndice A.

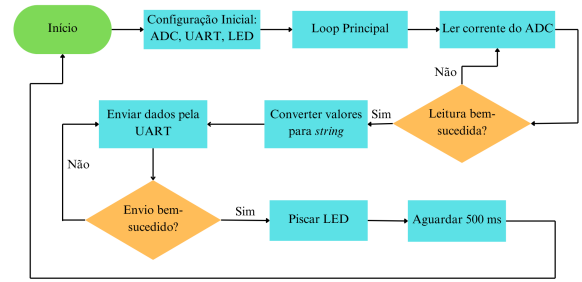



Figura 5: Fluxograma do projeto no MikroC.

Já a programação do ESP32 atuando como módulo Wifi foi realizada utilizando o *software* Arduino IDE, com o fluxograma do projeto observado na Figura 6. O módulo wifi recebe os dados de corrente do PIC16f877A via UART e os dados recebidos são convertidos em valores numéricos. Após isso, os dados são enviados para o Firebase usando o JSON, onde um LED sinaliza o envio com sucesso dos dados. O Apêndice A mostra o trecho do código.

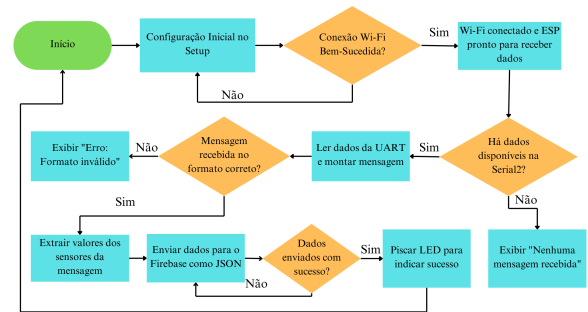



Figura 6: Fluxograma do projeto na IDE Arduino.

1) Aplicativo e *WebPage*

O repositório disponível em github.com/Projeto4HDAEE/Projeto4HDAEE apresenta o desenvolvimento de todo o sistema integrado composto por um aplicativo móvel e uma página *web* para o monitoramento e análise do consumo energético no laboratório GPMSE do prédio 4H do DAEE. Este projeto foi desenvolvido com foco em acessibilidade, praticidade e uso de tecnologias modernas para a coleta e exibição de dados em tempo real.

O desenvolvimento foi organizado em duas principais frentes: a criação de uma interface *web* e o desenvolvimento de um aplicativo móvel. A interface *web* foi construída utilizando as linguagens HTML, CSS e JavaScript, visando oferecer uma visualização clara e intuitiva dos dados monitorados. Para a sincronização dos dados em tempo real, foi utilizado o *Firebase*, que possibilita o armazenamento e a recuperação rápida e eficiente de informações.

No âmbito do aplicativo móvel, a aplicação foi desenvolvida em Android *Studio*, com integração direta ao *Firebase* para sincronizar os dados coletados e permitir acesso remoto

aos usuários. O aplicativo foi projetado para oferecer uma interface simples e funcional, proporcionando ao usuário uma experiência prática e amigável para o acompanhamento dos dados de consumo energético.

O código-fonte do projeto foi documentado e versionado no *GitHub*, permitindo transparência no processo de desenvolvimento e facilidade para futuras melhorias e adaptações. A estrutura do repositório foi organizada de forma a separar os componentes principais do projeto, como os arquivos da interface *web*, os *scripts* para integração com o *Firebase*, e os códigos responsáveis pelo funcionamento do aplicativo móvel.

Este projeto demonstra a capacidade de integrar tecnologias modernas para resolver problemas específicos, como o monitoramento energético, oferecendo soluções acessíveis e personalizadas para os usuários.

a: Página Inicial (*Home*)

A página inicial do *website* apresenta uma interface limpa e organizada, destacando os laboratórios envolvidos no projeto: Laboratório Roque, Laboratório Bardo e Laboratório Vitor. Cada seção fornece informações relevantes sobre as atividades e objetivos de cada laboratório, permitindo ao usuário compreender o escopo do projeto, na Figura 7 é possível visualizar a pagina inicial.

Figura 7: Pagina inicial do *Website*

b: Página de Monitoramento em Tempo Real

Na Figura 8 é possível ver a página de monitoramento em tempo real que oferece aos usuários a capacidade de visualizar dados atualizados instantaneamente. Por meio de gráficos interativos e indicadores visuais, é possível acompanhar métricas como consumo de energia total, consumo de energia por cada fase monitorada, status dos dispositivos, gráfico histórico e outras variáveis relevantes. Essa funcionalidade é essencial para a análise contínua e tomada de decisões, garantindo eficiência no gerenciamento dos recursos ativos monitorados.

Figura 8: Pagina do painel de consumo do laboratório GPMSE

Ao clicar no botão lateral "Menu"é possível visualizar a pagina calendário que possibilita visualizar dados salvos dos dias de amostragem conforme a Figura 9. Também

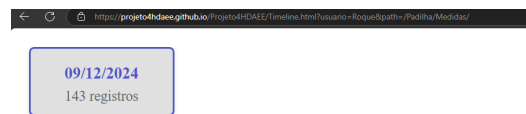



Figura 9: Pagina calendário de amostragem.

é possível visualizar o gráfico gerado por tal período de amostragem a Figura 10 demonstra o dia 09/12/2024.

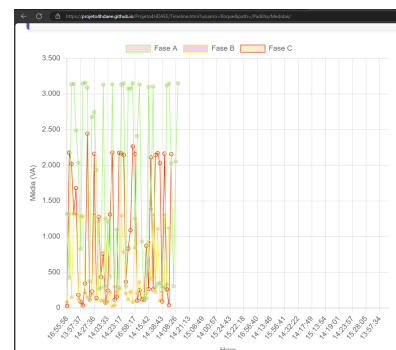



Figura 10: Pagina do gráfico do dia de registro selecionado.

D. COMPRA DE MATERIAIS UTILIZADOS

Para este projeto, foram comprados para utilização os seguintes componentes:

- 1 microcontrolador PIC-16F877A;
- 3 sensores de corrente STC-013;
- 1 microcontrolador ESP32;
- 2 capacitores de 22pF;
- Resistores;
- Cristal Oscilador;
- *Protoboard*;
- *Jumpers*.

E. TESTE DE COMPONENTES

Nesta etapa foram realizados testes simples para verificar o funcionamento dos componentes, como no microcon-

trolador PIC16F877A para verificar se estava em plena operação, além dos sensores de corrente. Para a verificação dos sensores STC, foi verificada a leitura da saída do TC, onde observou-se que o equipamento não apresentou nenhum defeito.

Os testes foram realizados também, nos resistores, capacitores e *jumpers* para o protótipo inicial da *protoboard*, para posteriormente, implementar na placa de circuito impresso (PCB).

Para a análise da medição dos laboratórios, foi utilizado como estudo, o laboratório do Grupo de Pesquisa em modelagem de sistemas elétricos (GPMSE). O GPMSE possui uma carga total de 20270VA com uma distribuição de montagens vista na Tabela 2.

Local	Tensão	Potência Total	Pot. Fase A	Pot. Fase B	Pot. Fase C
GPMSE	127V/220V	20270VA	7357VA	6924VA	5989VA

Tabela 2: Características do laboratório do Grupo de pesquisa em modelagem de sistemas elétricos.

F. MONTAGEM DO CIRCUITO

Primeiramente, a *protoboard* foi alimentada com 5V e com o *ground* do ESP32. Em seguida, o PIC16F877A foi alimentado com 5V e com o *ground* através das entradas VSS e VDD. Após isso, a saída TX do ESP32 foi conectada à entrada RX do PIC16F877A, enquanto a saída TX do PIC16F877A foi conectada à entrada RX do ESP32 por meio de um divisor de tensão com resistores de 2,2kΩ e 3,3kΩ, a fim de reduzir o sinal de 5V para 3V. Na sequência, utilizou-se um cristal oscilador de 16 MHz entre os pinos 13 e 14 do PIC16F877A. Do pino 13 para o terra, foi utilizado um capacitor de 22pF, assim como do pino 14 para o terra. Também foi utilizado um resistor de 10kΩ entre a alimentação de 5V e o pino de reset do PIC16F877A, denominado MCLR.

Para o funcionamento dos sensores, foi montado um circuito constituído por dois resistores de 10kΩ entre a alimentação de 5V e o *ground*, que configura um divisor de tensão, um capacitor de 10µF em paralelo com o resistor inferior, e um resistor de 33kΩ, conectado entre o divisor de tensão e um novo nó do circuito. O sensor de corrente STC-013 foi ligado em paralelo com o resistor de 33kΩ, sendo o cabo vermelho conectado ao divisor de tensão e o cabo branco ao nó direcionado ao PIC16F877A.

Para cada sensor, foi montado o circuito descrito anteriormente, e cada sensor foi conectado, respectivamente, às entradas RA0, RA1 e RA2 do PIC16F877A, que lê sinais analógicos e os converte internamente em sinais digitais. A Figura 11 apresenta o circuito montado.

Figura 11: Montagem do Circuito.

G. CORREÇÃO DE ERROS

Nessa atividade, foram corrigidos erros, como a correção da programação, assim como a inclusão da conexão da rede para verificar o tempo de resposta para o envio da mensagem com o *Wi-Fi* da universidade. Foi observado que havia problemas quando os valores de corrente chegavam ao módulo *wi-fi*, onde foi realizada uma calibração para os valores lidos serem compatíveis com os dados enviados.

H. CÁLCULO DO ERRO

Para calcular o erro, foi utilizado como base a equação 1 e definido como margem máxima 5%:

$$Erro(\%) = \frac{(ValorSensor - ValorReal)}{ValorReal} \cdot 100 \quad (1)$$

I. MONTAGEM FINAL DO PROJETO

Após todas as etapas concluídas, foi montado o protótipo do projeto e uma placa de circuito impresso que foram inseridos próximos aos quadros de distribuição do Laboratório do Grupo de Pesquisa em Modelagem de Sistemas Elétricos (403-4H) e do laboratório de Controle e Sistemas Microprocessados (Sala 404-4H). A Figura 12 mostra o circuito finalizado.

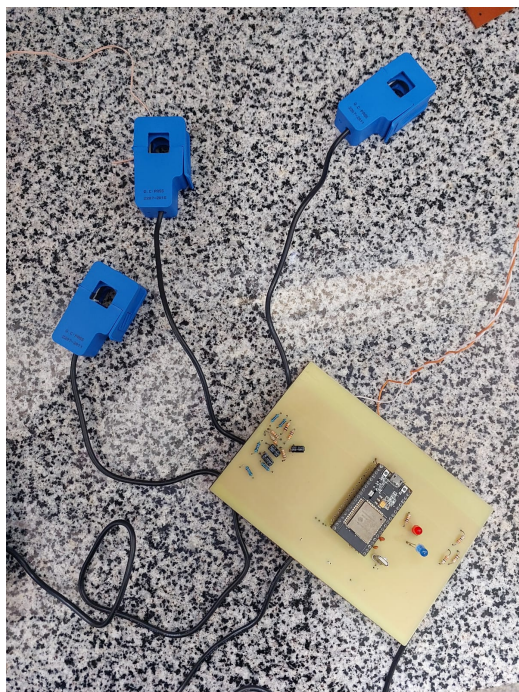



Figura 12: Montagem do Circuito.

V. CRONOGRAMA DE ATIVIDADES

As atividades desenvolvidas são apresentadas na Tabela 3.

Atividades	set	set	out	out	nov	nov
A	x					
B	x	x				
C	x	x				
D		x				
E			x			
F			x	x		
G				x	x	
H						x

Tabela 3: Cronograma de atividades.

VI. ALOCAÇÃO DE ATIVIDADES

As atividades foram alocadas para cada membro de acordo com a Tabela 4.

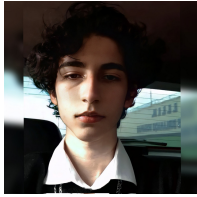
Atividades	set	set	out	out	nov	nov
Guilherme	A,B	B,C,D	E,F	F,G	G	H
Padilha	A,B	B,C,D	E,F	F,G	G	H

Tabela 4: Alocação de tarefas.

VII. RESULTADOS OBTIDOS

Os resultados obtidos com o projeto de monitoramento do consumo energético do laboratório do Grupo de Pesquisa em Modelagens de Sistemas Elétricos confirmaram a eficácia do sistema desenvolvido. Utilizou-se sensores não invasivos STC-013, o microcontrolador ESP32 para comunicação *Wi-Fi* e a plataforma *Firebase* para armazenamento e visualização dos dados, foi possível monitorar o consumo de energia em tempo real. A precisão dos sensores variou de acordo com a corrente medida, no qual indicou erros maiores para correntes baixas (aproximadamente 5%) e erros menores à medida que a corrente aumentava, com a valores abaixo de 1% para correntes mais altas. Os dados de consumo de cada fase foram processados e enviados para a plataforma *Firebase*, permitindo a visualização em gráficos interativos, estes dados estão presentes na Tabela 5 do Apêndice A.

Esse sistema mostrou-se eficiente não apenas na medição do consumo em cada uma das três fases, mas também na transmissão dos dados para a nuvem, garantindo o acompanhamento remoto do consumo energético. O uso do *Firebase* possibilitou a exibição dos dados em tempo real, que facilita a análise do comportamento energético do laboratório. Com base nos resultados obtidos, pode-se afirmar que a solução proposta é precisa e escalável, podendo ser adaptada para monitoramento de consumo energético em outros ambientes ou sistemas, oferecendo uma ferramenta útil para otimização do uso de energia.

Embora os valores anotados e os erros apresentados pelos sensores STC-013 tenham sido compatíveis com os objetivos do projeto, é possível aprimorar a precisão das medições através da coleta de uma sequência de variações para cada medição, de modo a obter um valor médio mais preciso. Esse procedimento é comumente utilizado em sensores de análise de qualidade de energia, onde múltiplas amostras são coletadas ao longo do tempo e a média das medições é calculada para minimizar os efeitos de flutuações momentâneas ou interferências. Implementar esse tipo de estratégia no sistema de monitoramento proposto pode resultar em uma maior precisão nas medições de corrente e consumo, especialmente para correntes mais baixas, e seria uma melhoria a ser considerada para trabalhos futuros. Além disso, tal abordagem permitiria validar de forma mais robusta os dados obtidos e garantir maior confiabilidade nas leituras em tempo real.

GUILHERME ROQUE ALMEIDA DE SOUSA. Aluno vinculado a Universidade Federal de Rondônia - UNIR, sob a matrícula 20202002057, no curso de ENGENHARIA ELÉTRICA - PORTO VELHO - BACHARELADO - Presencial - RO. Nascido em 2002 na cidade de Porto Velho, RONDÔNIA.

JOÃO VITOR DOS SANTOS PADILHA. Discente vinculado a Universidade Federal de Rondônia - UNIR, sob a matrícula 20202003377, no curso de ENGENHARIA ELÉTRICA - PORTO VELHO - BACHARELADO - Presencial - RO. Nascido em 2001 na cidade de Porto Velho, RONDÔNIA.

Referências

[1] MOGAJI, Emmanuel; ADEGBITE, Adetunji Adekunle. *Energy management in academic institutions: A review and proposed framework for energy efficiency. Energy Reports*, v. 7, p. 157-165, 2021.

[2] JAVED, Abdul Basit; SHIRAZ, Muhammad; HASHIM, Muhammad. *Design and implementation of IoT-based energy monitoring system using Firebase. International Journal of Smart Home*, v. 14, n. 2, p. 33-50, 2022.

[3] PEREIRA, João; DIAS, Luís; OLIVEIRA, Armando. *IoT-based energy monitoring: A comprehensive review of challenges and opportunities. IEEE Transactions on Industry Applications,* v. 56, n. 5, p. 4532-4543, 2020.

[4] ESPRESSIF SYSTEMS. ESP32-WROOM-32 *Datasheet*. Disponível em: https://www.espressif.com/en/products/modules/esp32. Acesso em: 01 dez. 2024.

[5] MICROCHIP TECHNOLOGY INC. PIC16F877A *Datasheet*. Disponível em: https://www.microchip.com/en-us/product/PIC16F877A. Acesso em: 01 dez. 2024.

[6] BEIJING YAOHUADECHANG ELECTRONIC CO. LTD. *Datasheet*: SCT-013-000. Disponível em: https://www.yhdc.com. Acesso em: [12/12/2024].

[7] MANSANI, Gabriel Lima. **Avaliação da Qualidade de Energia Elétrica do Prédio de Engenharia Elétrica da Universidade Federal de Rondônia**. Porto Velho, Universidade Federal de Rondônia, 2021. Disponível em: https://ri.unir.br/jspui/bitstream/123456789/4753/1/TCC%20-%20Gabriel%20Lima%20Mansani.pdf. Acesso em: 17 set. 2024.

APÊNDICE A- TRECHO DE CÓDIGO

```
// Configuração dos Fuses
#pragma config FOSC = HS        // Oscilador externo de alta frequência
#pragma config WDTE = OFF       // Watchdog Timer desabilitado
#pragma config PWRTE = ON       // Power-up Timer habilitado
#pragma config BOREN = ON       // Brown-out Reset habilitado
#pragma config LVP = OFF        // Baixa tensão de programação desabilitada
#pragma config CPD = OFF        // Proteção de memória de dados desabilitada
#pragma config WRT = OFF        // Proteção de memória de programa desabilitada
#pragma config CP = OFF         // Proteção de código desabilitada

#define _XTAL_FREQ 16000000 // 16 MHz

// Definição do LED
sbit LED_PIN at RD0_bit;            // Pino do LED
sbit LED_PIN_DIRECTION at TRISD0_bit; // Direção do pino do LED

// Protótipos de funções
void setup();
float LeCorrente(unsigned short nIn);

void setup() {
    CMCON = 0x07;                   // Desativa os comparadores
    ADCON1 = 0b1000;                // Configuração ADC: VSS/GND como Vref-, VDD como Vref+, AN0-AN3 habilitados
    ADCON0 = 0b00000001;            // Habilita o ADC, seleciona o canal AN0

    LED_PIN_DIRECTION = 0;      // Configura RD0 como saída
    LED_PIN = 0;                    // Inicializa o LED desligado

    UART1_Init(9600);               // Inicializa UART com baud rate de 9600 bps
    Delay_ms(100);                  // Aguarda estabilização da UART
}

void main() {
    float corrente[3];
    char buffer[16];   // Buffer para montagem de strings
    unsigned short i;

    setup();

    while (1) {
        // Leitura das correntes
        for (i = 0; i < 3; i++) {
            corrente[i] = LeCorrente(i + 1);
        }

        // Envia os valores pela UART em uma única mensagem
        for (i = 0; i < 3; i++) {
            //long parteInteira = (long)corrente[i];  // Parte inteira
            //long parteDecimal = (long)((corrente[i] - parteInteira) * 1000);  // Parte decimal com 3
```

```
                  casas

            FloatToStr(corrente[i], buffer);              // Converte a parte inteira para string
            UART1_Write_Text(buffer);                      // Envia a parte inteira
            //UART1_Write('.');                            // Envia o ponto decimal
            //IntToStr(parteDecimal, buffer);              // Converte a parte decimal para string
            //UART1_Write_Text(buffer);                    // Envia a parte decimal

            if (i < 2) {
                UART1_Write(';');                          // Adiciona delimitador entre valores
            }
        }

        UART1_Write('\n');  // Finaliza a mensagem com nova linha

        // Pisca o LED para indicar envio
        LED_PIN = 1;
        Delay_ms(200);
        LED_PIN = 0;

        Delay_ms(500);  // Atraso para evitar saturação
    }
}

float LeCorrente(unsigned short nIn) {
    float TEMP[2] = {0, 1024};
    float PCMVal;
    float RelVolt, Ampere;
    unsigned short i, iterations = 254;  // Aumenta iterações para maior precisão

    for (i = 0; i < iterations; i++) {
        PCMVal = ADC_Read(nIn - 1); // Leitura do ADC
        if (TEMP[0] < PCMVal) TEMP[0] = PCMVal;
        if (TEMP[1] > PCMVal) TEMP[1] = PCMVal;
    }

    RelVolt = (float)(TEMP[0] - TEMP[1]) * 0.13808; // Conversão para mV
    Ampere = (RelVolt * 0.707);            // Valor RMS em amperes

    return (Ampere < 0) ? -Ampere : Ampere;  // Retorna valor absoluto
```

Listing 1: Trecho de código no MikroC.

```
#include <Firebase.h>

#define FIREBASE_HOST "consumo4hdaee-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "noG7SmONxUnWvvyTcOAmZAlnTISDtWC6JuWVABbv"

#define WIFI_SSID "UNIR"
#define WIFI_PASSWORD "r1c1unir"

#define LED 18
Firebase firebase(FIREBASE_HOST, FIREBASE_AUTH);
FirebaseData fbdo;
void setup() {
  Serial.begin(9600);                              // Inicializa a comunicação serial para o Monitor Serial
  Serial2.begin(9600, SERIAL_8N1, 16, 17);   // Configura Serial2 com RX=16, TX=17
  pinMode(LED, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.println("Conectando ao Wi-Fi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWi-Fi conectado!");
  Serial.println("ESP32 pronto para receber dados...");
}

void loop() {
  if (Serial2.available() > 0) {  // Verifica se há dados disponíveis na UART
    String receivedMessage = "";  // Variável para armazenar a mensagem recebida

    while (Serial2.available() > 0) {        // Lê enquanto há caracteres disponíveis
      char receivedChar = Serial2.read();  // Lê um caractere da UART
      receivedMessage += receivedChar;       // Adiciona o caractere  mensagem
      delay(2);                              // Aguarda para garantir a recepção completa de strings curtas
    }

    // Exibe a mensagem recebida no Monitor Serial
    Serial.print("Mensagem recebida: ");
    Serial.println(receivedMessage);

    // Remove espaços extras
    receivedMessage.replace(" ", "");   // Remove todos os espaços na mensagem

    // Processa a mensagem recebida
    float sensor1 = 0.0, sensor2 = 0.0, sensor3 = 0.0;
    int firstSeparator = receivedMessage.indexOf(';');                    // Localiza o primeiro separador
    int secondSeparator = receivedMessage.indexOf(';', firstSeparator + 1);  // Localiza o segundo separador

    if (firstSeparator != -1 && secondSeparator != -1) {  // Verifica se os separadores existem
      // Extrai os valores correspondentes a cada sensor
      sensor1 = receivedMessage.substring(0, firstSeparator).toFloat();
      sensor2 = receivedMessage.substring(firstSeparator + 1, secondSeparator).toFloat();
      sensor3 = receivedMessage.substring(secondSeparator + 1).toFloat();
      FirebaseJson dados;

```

```
56        // Exibe os valores de cada sensor no Monitor Serial com três casas decimais
57        Serial.print("Sensor 1: ");
58        Serial.println(sensor1, 3);  // Exibe com 3 casas decimais
59        dados.setFloat("CorrenteA", sensor1);
60        digitalWrite(LED, HIGH);  // Liga o LED
61        delay(10);                // Aguarda 100 ms
62        digitalWrite(LED, LOW);   // Desliga o LED
63        delay(10);                // Aguarda 100 ms
64
65        Serial.print("Sensor 2: ");
66        Serial.println(sensor2, 3);  // Exibe com 3 casas decimais
67        dados.setFloat("CorrenteB", sensor2);
68        digitalWrite(LED, HIGH);  // Liga o LED
69        delay(10);                // Aguarda 100 ms
70        digitalWrite(LED, LOW);   // Desliga o LED
71        delay(10);                // Aguarda 100 ms
72
73        Serial.print("Sensor 3: ");
74        Serial.println(sensor3, 3);  // Exibe com 3 casas decimais
75        dados.setFloat("CorrenteC", sensor3);
76        digitalWrite(LED, HIGH);  // Liga o LED
77        delay(10);               // Aguarda 100 ms
78        digitalWrite(LED, LOW);   // Desliga o LED
79        delay(10);               // Aguarda 100 ms
80        digitalWrite(LED, HIGH);  // Liga o LED
81        delay(10);               // Aguarda 100 ms
82        digitalWrite(LED, LOW);   // Desliga o LED
83        delay(10);               // Aguarda 100 ms
84        Firebase.setJSON(fbdo, "BARDO/Medidas/", dados);
85        Serial.println("Dados enviados para o Firebase!");
86      } else {
87        Serial.println("Erro: Formato de mensagem inválido.");
88      }
89
90    } else {
91      Serial.println("Nenhuma mensagem recebida"); // Informação de debug
92    }
93 }
```

Listing 2: Trecho de código na IDE do Arduino.

Fase	Corrente Real (A)	Corrente Sensor (A)	VA Sensor (VA)	Erro (%)
A	0.532	0.505	64.2	5.0%
A	1.256	1.212	154.8	3.5%
A	2.124	2.080	265.6	2.1%
A	3.654	3.586	455.7	1.9%
A	4.912	4.844	617.1	1.4%
A	6.043	5.921	756.7	2.0%
A	8.067	7.883	1009.5	2.3%
A	9.542	9.418	1194.8	1.3%
A	11.226	10.948	1411.5	1.9%
A	13.001	12.861	1635.0	1.1%
B	0.654	0.615	77.9	5.9%
B	1.541	1.491	189.1	3.2%
B	2.349	2.291	291.5	2.5%
B	3.209	3.143	405.4	2.1%
B	4.511	4.457	566.2	1.2%
B	5.472	5.397	689.9	1.4%
B	7.036	6.982	885.0	0.8%
B	8.234	8.150	1017.3	1.0%
B	9.178	8.912	1144.0	2.3%
B	11.341	11.261	1424.2	0.7%
C	0.845	0.799	175.7	5.4%
C	1.732	1.686	369.2	2.6%
C	2.501	2.442	538.4	2.4%
C	3.687	3.614	796.2	2.0%
C	5.233	5.148	1132.5	1.6%
C	6.314	6.250	1378.9	1.0%
C	8.013	7.947	1746.5	0.8%
C	9.423	9.184	2032.2	1.3%
C	10.598	10.273	2307.7	1.7%
C	12.015	11.922	2610.9	0.8%

Tabela 5: Comparação entre os valores medidos e os valores reais indicados pelo sensor para as fases A, B e C